

REMARKS

[0001] Applicant respectfully requests reconsideration and allowance of all of the claims of the application. Claims 1-5, and 18-24 are presently pending. Claims 1, 4, and 18 are amended for a minor grammatical errors.

Formal Request for an Interview

[0002] If the Examiner's reply to this communication is anything other than allowance of all pending claims, then I formally request an interview with the Examiner. I encourage the Examiner to call me—the undersigned representative for the Applicant—so that we can talk about this matter so as to resolve any outstanding issues quickly and efficiently over the phone.

[0003] Please contact me to schedule a date and time for a telephone interview that is most convenient for both of us. While email works great for me, I welcome your call as well. My contact information may be found on the last page of this response.

Claim Rejections under § 102

[0004] The Examiner rejects claims 1-5 and 18-24 under § 102. For the reasons set forth below, the Examiner has not shown that the cited reference anticipates the rejected claims.

[0005] Accordingly, Applicant's attorney respectfully requests that the § 102 rejections be withdrawn and the case be passed along to issuance.

[0006] The Examiner's rejections are based upon the following references alone:

- **US Patent No. 7,346,780 to Sinha et al:** "*Sinha et al*" hereinafter, (issued October 9, 2003).

Anticipation Rejections

[0007] Applicant's attorney submits that the anticipation rejections are not valid because, for each rejected claim, no single reference discloses each and every element of that rejected claim.¹ Furthermore, the elements disclosed in the single reference are not arranged in the manner recited by each rejected claim.²

Based upon *Sinha et al*

[0008] The Examiner rejects claims 1-5 and 18-24 under 35 U.S.C. § 102(e) as being anticipated by *Sinha et al*. Applicant's attorney respectfully traverses the rejection of these claims. Based on the reasons given below, Applicant's attorney asks the Examiner to withdraw the rejection of these claims.

Independent Claim 1

[0009] Applicant's attorney submits that *Sinha et al* does not anticipate this claim because it does not disclose all of the elements as recited in this claim. In specific, claim 1 recites calculating, on each task change between a first program module and a second program module of the CPU, a signature of at least part of the program instruction lines, and checking the conformity of this signature with a signature recorded upon previous execution of the involved program.

¹ "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987); also see MPEP §2131.

² See *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

[0010] For example, according to an embodiment, a CPU may have the capability of multitasking between executing two or more separate and distinct program modules. As one program module is executed, it may be referred to as running in the foreground. Other program modules that are executing, but are not currently engaged by the CPU in the execution of specific tasks may be referred to as running in the background. Thus, in an effort to ensure that a background program module has not been tampered with, each time a program module transitions to become a foreground application in which the CPU is executing instructions from said application, at least part of the application is used to generate a signature. This signature may be compared to a stored signature that was generated and stored in a memory when the application transitioned to the background. If the signatures match, then this match may be interpreted to mean that the program module now transitioning back to the foreground has not been changed or modified. See generally, paragraph [8] of the detailed description section of the specification.

[0011] However, *Sinha et al* teaches a method that has several differences. Ultimately, *Sinha et al* does not teach the recitations of claim 1. In specific, *Sinha et al* is directed, generally, toward a digital rights management (DRM) solution that may verify that two instances of the same program module are unaltered duplicates of each other, using a so-called “integrity veracitication” technique as generally described in the cited and applied section, column 7, line 31 to column 8, line 25 of *Sinha et al*. This section of *Sinha et al* checks multiple serial execution instances of a single program module against each other. Thus, an output tracing method may check an execution-identifying signature each time

the one program module is executed to verify that the signature matches other signatures stored at previous execution times.

[0012] However, *Sinha et al* offers no teaching or discussion about performing such a signature calculation in response to a CPU switching tasks between two different and simultaneously executing programs. Claim 1 recites calculating, on each task change between a first program module and a second program module of the CPU, a signature. The teachings of *Sinha et al* necessarily assume that each execution of an application is a self-contained event in that *Sinha et al* does not contemplate a situation wherein the execution of an application may be interrupted when the application is pushed from a foreground execution to a background execution by the CPU. Moreover, there is no teaching anywhere in *Sinha et al* with regard to a multitask CPU as also recited in claim 1.

[0013] Consequently, *Sinha et al* does not disclose all of the elements and features of this claim. Accordingly, Applicant's attorney asks the Examiner to withdraw the rejection of this claim.

Dependent Claims 2-3

[0014] These claims ultimately depend upon independent claim 1. As discussed above, claim 1 is allowable. It is axiomatic that any dependent claim which depends from an allowable base claim is also allowable. Additionally, some or all of these claims may also be allowable for additional independent reasons.

Independent Claim 4

[0015] Applicant's attorney submits that *Sinha et al* does not teach all of the elements as recited in this claim. Claim 4 recites a processor of multitask execution of several programs, exploiting a table of correspondence, each correspondence being associated with an identifier of the involved program, comprising means for calculating a current signature, and means for comparing this signature with the identifier of the program stored in the correspondence table.

[0016] Claim 4 is patentable for at least similar reasons as recited above in support of the patentability of claim 1. As shown above, *Sinha et al* does not teach all of the elements and features of this claim. Accordingly, Applicant's attorney asks the Examiner to withdraw the rejection of this claim.

Dependent Claim 5

[0017] This claim ultimately depends upon independent claim 4. As discussed above, claim 4 is allowable. It is axiomatic that any dependent claim which depends from an allowable base claim is also allowable. Additionally, this claim may also be allowable for additional independent reasons.

Independent Claim 18

[0018] Applicant's attorney submits that *Sinha et al* does not anticipate this claim because it does not disclose all of the elements as recited in this claim. In

specific, claim 18 recites calculating, on each task change, a new signature, and checking the conformity of the new signature with the unique signature.

[0019] As was exemplified above, according to an embodiment, a CPU may have the capability of multitasking between executing two or more separate and distinct program modules that are executing at the same CPU. In an effort to ensure that a background program module has not been tampered with, each time a program module transitions to become a foreground application in which the CPU is executing instructions from said application, at least part of the application is used to generate a new signature. This new signature may be compared to a unique signature that was generated and previously stored in a memory when the application transitioned to the background. If the signatures match, then this match may be interpreted to mean that the program module now transitioning back to the foreground has not been changed or modified. See generally, paragraph [8] of the detailed description section of the specification.

[0020] As detailed above, *Sinha et al* teaches a method that has several differences. In specific, *Sinha et al* is teaches, as generally described in the cited and applied section, column 7, line 31 to column 8, line 25 of *Sinha et al*, a method for checking multiple execution instances of a single program module against each other. There is simply no disclosure in *Sinha et al* that teaches that any program modules is executing simultaneously with any other program module or that there are, in fact, two different and distinct program modules. Rather, an output tracing method may check an execution-identifying signature each time a program module is executed to verify that the signature matches other signatures stored at previous execution times.

[0021] Further, *Sinha et al* offers no teaching or discussion about performing such a signature calculation in response to a CPU switching tasks. Claim 18 recites calculating, on each task change, a new signature. The teachings of *Sinha et al* necessarily assume that each execution of an application is a self-contained event in that *Sinha et al* does not contemplate a situation wherein the execution of an application may be interrupted when the application is pushed from a foreground execution to a background execution by the CPU. Moreover, there is no teaching anywhere in *Sinha et al* with regard to a multitask CPU as also recited in claim 18.

[0022] Consequently, *Sinha et al* does not disclose all of the elements and features of this claim. Accordingly, Applicant's attorney asks the Examiner to withdraw the rejection of this claim.

Dependent Claims 19-22

[0023] These claims ultimately depend upon independent claim 18. As discussed above, claim 18 is allowable. It is axiomatic that any dependent claim which depends from an allowable base claim is also allowable. Additionally, some or all of these claims may also be allowable for additional independent reasons.

[0024] For example, claim 21 recites checking the conformity of a next new signature with the stored new signature at the next task change associated with the program. *Sinha et al* does not teach anything with respect to a CPU capable of multitasking and, therefore, cannot possibly be construed to teach a specific

nature of multitasking as recited in claim 21. For at least this additional reason, Applicant's attorney submits that claim 21 is allowable over the prior art of record.

Conclusion

[0025] All pending claims are in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the application. If any issues remain that prevent issuance of this application, the **Examiner is urged to contact me before issuing a subsequent Action**. Please call or email me at your convenience.

[0026] Any additional fees required as a result of this amendment have been paid from the below-referenced deposit account as filed herewith. Should further payment be required to cover such fees you are hereby authorized to charge such payment to Deposit Account No. 07-1897.

Respectfully Submitted,

Graybeal, Jackson LLP
Representatives for Applicant

/Kevin D. Jablonski/
Kevin D. Jablonski (kevin@graybeal.com)
Registration No. 50,401
USPTO Customer No.: 00996

Dated: October 13, 2009

Telephone: (425) 455-5575
Facsimile: (425) 455-1046